CODE DESIGN FOR FAST SELECTIVE RETRIEVAL OF FUSION STORED SENSOR-NETWORK/TIME-SERIES DATA

Sharadh Ramaswamy, Jayanth Nayak† and Kenneth Rose**

* ECE Department, University of California, Santa Barbara, CA 93106, USA.
† EE Department, University of California, Riverside, CA 92521, USA.
{rsharadh,rose}@ece.ucsb.edu, jnayak@ee.ucr.edu

ABSTRACT

We consider the problem of storing data from multiple correlated sources in a database, so as to enable efficient future retrieval of data from select subsets of the sources, where only statistical information about queries may be available in advance. This setting poses new challenges in terms of the precise tradeoffs between storage rate, retrieval rate and distortion. We derive a gradient descent algorithm to optimize the joint encoding (storage) procedure, the decoder, and the retrieval procedure via mapping from queries to subsets of the stored data to retrieve and decode, so that the average retrieval rate-distortion cost is minimized, given a pre-specified overall storage capacity (or rate). Experiments conducted on real and synthetic data-sets demonstrate that our selective retrieval procedure is able to achieve significantly better trade-offs than joint compression, with retrieval speed-ups reaching 5X and distortion gains of up to 3.5dB possible.

Index Terms— Multisensor systems, Vector quantization, Data compression, Database query processing

1. INTRODUCTION

We are motivated by the problem of data storage for sensor networks. As an illustrative example, suppose we consider a network of surveillance cameras covering a scene. The video signals generated by these cameras are expected to be highly correlated, since they are covering the same scene. This data is sent to a fusion center to be stored for possible future analysis. In a slightly more generalized setting, we could possibly have *multiple* storage centers each of which store video data from one or more cameras i.e. *distributed storage* but for the sake of simplicity, we assume that all video signals are stored in a single fusion center.

When the data from the fusion center is eventually accessed by a user, it is very likely that the views from only a small subset, and not all, of the cameras will be requested at any given time. An interesting tradeoff emerges between conflicting objectives: On the one hand, the inter-sensor correlation may be exploited via joint coding to minimize the overall storage requirement and to minimize the retrieval bit rate (and hence time) required for retrieving highly correlated data. On the other hand, the specific nature of the query may result in selecting only very few of the sources to be reconstructed, and it would be wasteful to have to retrieve the entire compressed data only to reconstruct a small subset. Figure 1 is representative of the issues at hand.

The work is supported in part by the NSF under grant IIS-0329267, the University of California MICRO program, Applied Signal Technology, Inc., Dolby Laboratories Inc., and Qualcomm Inc.

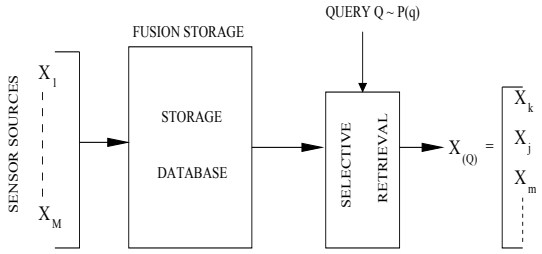

Fig. 1. Fusion Storage vs. Selective Retrieval

The high level of correlation between signals generated by sensors in a network has been exploited to minimize the requirements on the communication link from sensors to the fusion center (or collector node) [1]. But to the best of our knowledge, no work exists to model, let alone solve, the constrained optimization problem of *efficient storage* and *fast retrieval* of data generated by sensor networks. We would like to emphasize here that while we are motivated by sensor networks, the problem *generalizes to the storage of all collections of correlated sources/time-series*. We have recently pointed to the underlying theoretical problem in [2], where we also provided information-theoretic (asymptotic) analysis to determine an achievable rate region for lossless storage and reconstruction. In subsequent sections, we first show the non-trivial nature of the problem (section 2), then describe our solution framework (section 3) and the design algorithm (section 4), (that optimizes the tradeoff between storage capacity, distortion, and retrieval speed) and finally, report experimental results (section 5).

2. INFORMATION THEORETIC MOTIVATION

Let us denote the M correlated sources as the set, $\{X_m, m = 1...M\}$. We define a query as the subset of sources that need to be retrieved. Employing binary variables $q_i \in \{0, 1\}$ to denote whether source X_i is requested or not, we represent queries by M-tuples of the form

$$\mathbf{q} = (q_1, ..., q_M) \in \mathcal{Q} \tag{1}$$

where $\mathcal{Q} \subseteq \{0, 1\}^M$ represents the domain-set of queries. We next introduce notation for the query distribution, or the probability mass function (pmf),

$$P : \mathcal{Q} \rightarrow [0, 1] \tag{2}$$

It is to be noted that there are conceivably 2^M possible queries and $\sum_{\mathbf{q}\in\mathcal{Q}} P(\mathbf{q}) = 1$. Without loss of generality, we assume that each source is requested with positive probability (i.e., there exists some

query with positive probability whose requested subset includes the source) and that a query always asks for a non-empty subset of sources, i.e.,

$$P(\mathbf{0}) = 0 \tag{3}$$

It is to be noted that in our notation, boldface letters in lowercase and upper case represent vectors and random vectors, respectively.

Given a database of constant size, the *retrieval time* or the time required to retrieve a subset of sources is proportional to the number of bits retrieved *per sample*, which we term the *retrieval rate*. Let the number of bits pulled out to answer query $\mathbf{q}$ be $R_{\mathbf{q}}$. Then, the average retrieval rate is

$$R_r = \sum_{\mathbf{q}\in\mathcal{Q}} P(\mathbf{q}) R_{\mathbf{q}} \tag{4}$$

Since *retrieval speed* is inversely proportional to *retrieval time*, our goal of *maximizing the retrieval speed* is trivially equivalent to *minimizing the retrieval rate*.

2.1. Lossless Storage and Retrieval

To compress any source of information X, the number of bits required for lossless representation should be at least the Shannon entropy (*entropy-rate*, for sources with memory and *differential entropy*, for continuous alphabets) of the source, $H(X)$.

2.1.1. Minimal Storage Rate

It follows from Shannon's basic result that the minimum number of bits required to store M sources $X_1, \ldots, X_M$, i.e. the minimal *storage rate* is

$$R_{s,min} = H(X_1, ..., X_M) \tag{5}$$

Since $H(X_1, ..., X_M) \leq \sum_m H(X_m)$ joint compression of correlated sources *always* requires less bits for storage than separate compression (by exploiting inter-source redundancies). It is also to be noted that the retrieval rate for this method is

$$R_r = H(X_1, ..., X_M) \tag{6}$$

since *the entire compressed description needs to be retrieved for any query*.

2.1.2. Minimal Retrieval Rate

If we denote the set of sources queried as,

$$X_{(\mathbf{q})} \quad = \quad \{X_m, \forall m : q_m = 1\}$$

the minimum number of bits required to reconstruct the sources requested in query $\mathbf{q}$ is $H(X_{(\mathbf{q})})$ and hence, the minimum average retrieval rate possible for any query distribution is

$$R_{r,min} = \sum_{\mathbf{q}} P(\mathbf{q}) H(X_{(\mathbf{q})}) \leq H(X_1, ..., X_M)$$

This implies that *joint compression is not optimal in retrieval speed*, and in order to have the fastest retrieval speed, we need to *compress and store each subset* of sources that may be requested, *separately*. However, unless M is very small or the set of queries $\mathcal{Q}$ is severely restricted, the storage requirement would be impractically high as it would have to individually accommodate a very large (possibly an *exponential*) number of queries, i.e.

$$R_s = \sum_{\mathbf{q}\in\mathcal{Q}} H(X_{(\mathbf{q})}) >> H(X_1, ..., X_M) = R_{s,min} \tag{7}$$

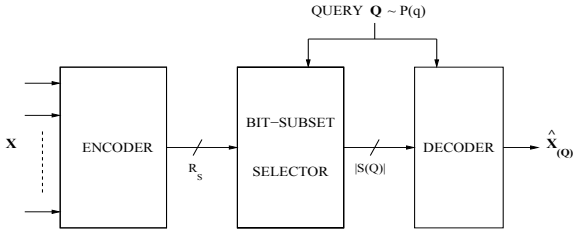

Fig. 2. Proposed System Block Diagram

Thus, it is clear that the optimum storage technique is wasteful in retrieval speed and the optimal retrieval technique is wasteful in storage.

3. LAGRANGIAN COST FORMULATION

Any practical storage scheme would need to quantize and compress the data before storage, hence leading to some error or *distortion*. The *reconstruction* distortion is measured as

$$d_{\mathbf{q}}(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{m=1}^{M} q_m d_m(x_m, \hat{x}_m), \tag{8}$$

where

$$d_m : \mathcal{R} \times \mathcal{R} \to [0, \infty). \tag{9}$$

Hereafter, we will specialize to the *squared error distortion* measure, i.e. distortion measure of the form

$$d_m(x, \hat{x}) = (x - \hat{x})^2, \qquad m = 1, ..., M \tag{10}$$

A block diagram, representative of our system model, is given in figure 2. We define the encoder as the function

$$\mathcal{E} : \mathcal{R}^M \to \mathcal{I} = \{0, 1\}^{R_s} \tag{11}$$

which compresses the M-dimensional input vector $\mathbf{x}$, representing the M sources, to R_s bits at each instant.

The bit subset selector is the mapping

$$\mathcal{S} : \mathcal{Q} \to \mathcal{B} = 2^{\{1,\ldots,R_s\}} \tag{12}$$

where $\mathcal{Q} \subseteq \{0, 1\}^M$ represents the domain-set of queries and $\mathcal{B}$ is power set (set of all subsets) of the set $\{1, ..., R_s\}$. This mapping determines which of the stored bits to retrieve for a given query $\mathbf{q}$. It is to be noted that $\mathcal{S}(\mathbf{q}) \subseteq \{1, \ldots, R_s\}, \forall \mathbf{q}$.

For each subset of bits $\mathbf{e}$ that can be retrieved, an estimate of all the sources is formed by the decoder

$$\mathcal{D} : \mathcal{I} \times \mathcal{B} \to \hat{\mathcal{X}} \tag{13}$$

where $\hat{\mathcal{X}} \subset \mathcal{R}^M$ is the corresponding codebook. The average distortion for a specific query $\mathbf{q}$ is

$$D_{\mathbf{q}} = E[d_{\mathbf{q}}(\mathbf{X}, \mathcal{D}(\mathcal{E}(\mathbf{X}), \mathcal{S}(\mathbf{q})))] \tag{14}$$

where $E[\ldots]$ denotes statistical expectation, and the distortion averaged across all queries is $D = \sum_{\mathbf{q}\in\mathcal{Q}} P(\mathbf{q}) D_{\mathbf{q}}$. In practice, since we have access to the database $\mathcal{X}$ itself, we use it as a training set and replace the expectation operator $E[\ldots]$ by a simple average, evaluated across the database $\mathcal{X}$. Hence, the distortion is evaluated as

$$D = \sum_{\mathbf{q}\in\mathcal{Q}} P(\mathbf{q}) \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} d_{\mathbf{q}}(\mathbf{x}, \hat{\mathbf{x}}), \tag{15}$$

Noting that $\mathcal{S}(\mathbf{q})$ denotes the subset of bits drawn to reconstruct query q and $R_{\mathbf{q}} = R_{\mathcal{S}(\mathbf{q})} = |\mathcal{S}(\mathbf{q})|$, the average retrieval rate is,

$$R_r = \sum_{\mathbf{q}} P(\mathbf{q}) R_{\mathbf{q}} = \sum_{\mathbf{q}} P(\mathbf{q}) |\mathcal{S}(\mathbf{q})| \quad (16)$$

Given M correlated sources, a storage constraint R_s and a distortion constraint D, we aim to design storage systems that

$$\min_{\mathcal{E},\mathcal{S},\mathcal{D}} J = R_r(R_s, D) \quad (17)$$

Reformulating the problem in terms of Lagrange multipliers, the objective is to

$$\min_{\mathcal{E},\mathcal{S},\mathcal{D}} J = D(R_s) + \lambda R_r(R_s), \lambda \geq 0 \quad (18)$$

4. NECESSARY CONDITIONS FOR OPTIMALITY

The Lagrangian cost J can be rewritten as

$$J = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}} \sum_{\mathbf{q}} P(\mathbf{q}) d_{\mathbf{q}}(\mathbf{x}, \mathcal{D}(\mathcal{E}(\mathbf{x}), \mathcal{S}(\mathbf{q}))) + \lambda \sum_{\mathbf{q}} P(\mathbf{q}) |\mathcal{S}(\mathbf{q})| \quad (19)$$

Optimal Encoder : From the above equation, the optimal encoding index $\mathcal{E}(\mathbf{x})$ for input vector $\mathbf{x}$ is

$$\mathcal{E}(\mathbf{x}) = \arg\min_{\mathbf{i} \in \mathcal{I}} \sum_{\mathbf{q}} P(\mathbf{q}) d_{\mathbf{q}}(\mathbf{x}, \mathcal{D}(\mathbf{i}, \mathcal{S}(\mathbf{q}))), \forall \mathbf{x}$$

Optimal Bit-subset Selector : Similarly, the best set of bits to be pulled out for a particular query should be the one that minimizes the Lagrangian sum of the distortion measure $d_{\mathbf{q}}(.,.)$, averaged across the training set, and the query-specific retrieval rate $R_{\mathbf{q}} = |\mathcal{S}(\mathbf{q})|$ i.e.

$$\mathcal{S}(\mathbf{q}) = \arg\min_{\mathbf{e} \in \mathcal{B}} \{ \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}} d_{\mathbf{q}}(\mathbf{x}, \mathcal{D}(\mathcal{E}(\mathbf{x}), \mathbf{e})) + \lambda |\mathbf{e}| \}, \forall \mathbf{q}$$

Optimal Decoder : We use $\mathbf{i}_{\mathbf{e}}$, to denote the sub-index extracted from $\mathbf{i}$ by retrieving the bits in the positions indicated by $\mathbf{e}$ and $\mathcal{D}(\mathbf{i}, \mathbf{e})$ to denote the corresponding codevector. By setting to zero, the partial derivatives of J w.r.t $\mathcal{D}(\mathbf{i}, \mathbf{e}) \forall \mathbf{e} \in \mathcal{B}$, we find the optimal decoder to be

$$\mathcal{D}(\mathbf{i}, \mathbf{e}) = \frac{1}{|F|} \sum_{\mathbf{x} \in F} \mathbf{x}, \forall \mathbf{e}, \mathbf{i}$$

where $F = \{\mathbf{x} : (\mathcal{E}(\mathbf{x}))_{\mathbf{e}} = (\mathbf{i})_{\mathbf{e}}\}$.

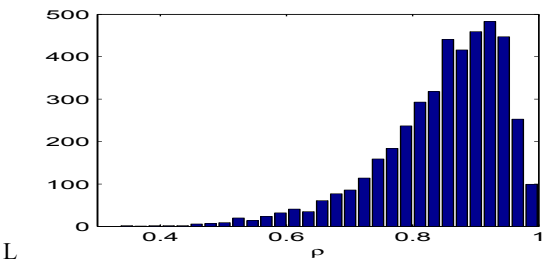

Fig. 3. STOCKS: Correlation histogram

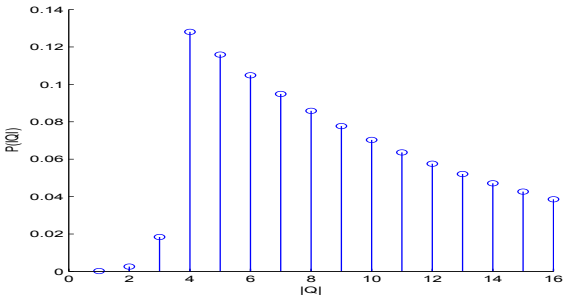

Fig. 4. EXP: Exponential Query Distribution with $M = 93$ sources

4.1. Algorithm for Lagrangian Cost Optimization

A natural algorithm is to iteratively enforce each of the necessary conditions for optimality (derived in the preceding sections), till a convergence condition is satisfied. In effect, the algorithm just partitions the elements of the training set and the storage bits into different groups, and for a finite sized training set and a finite storage rate, there exist only a finite number of set partitions. At each step of the optimization, a set of parameters are chosen to minimize the Lagrangian cost and hence, with every iteration, the cost is non-increasing. Therefore, the algorithm is *guaranteed* to *converge* in a finite number of iterations. It is to be noted that the Lagrangian cost surface is non-convex and has multiple local optima. Hence, a *globally optimal* solution is *not guaranteed*.

5. SIMULATION RESULTS

5.1. DATA-SETS

We tested our algorithm extensively on both synthetic and real data-sets, where we evaluated the *operational (retrieval) rate* (R_r) *vs. distortion* D *curves* for different settings of storage complexity. A brief description of our data-sets follows.

5.1.1. SYNTH: Synthetic data

For the synthetic data-set SYNTH, the sensor sources were modelled as correlated Gaussian sources (of unit variance i.e. $\sigma^2 = 1$), with the correlation between sources modelled as falling exponentially with distance. Specifically, if ρ_{ij} represents the correlation between sources X_i and X_j,

$$\rho_{ij} = \rho^{|i-j|} \quad (20)$$

where $-1 \leq \rho \leq 1$. This correlation model can be expected when spatio-temporal sensor fields are uniformly sampled [3]. We tested the system for $\rho = 0.3$ and $\rho = 0.8$, corresponding to low and high correlation data-sets , with $M = 93$ sources, each having 3000 samples.

5.1.2. STOCKS: Real Data

The real data-set that was used was the STOCKS data-set, available in the University of California, Riverside (UCR) Time-Series Data Mining Archive and consists of $M = 93$ stocks, each having 3000 samples. It is highly correlated, as can be seen from the histogram of the pairwise correlation coefficient ρ (Figure 3).

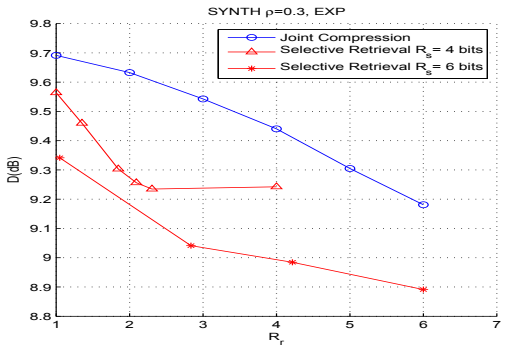

Fig. 5. Selective Retrieval vs. Joint Compression: SYNTH, $\rho = 0.3$

5.2. Query Model

In this section, we describe "*exponential queries*" ("EXP"), which we believe model real-user behavior. Even though in theory, there are $2^M - 1$ possible queries, typically, only a smaller number of sources (say n) are going to be requested at any time. There are $\binom{M}{n}$ ways of selecting n out of M objects and for moderate values of M, even this might be very large. For example, if $M = 30$ and $n = 4$, $2^{30} - 1 \simeq 10^9$and $\binom{30}{4} = 27405$. Intuitively, we believe that the typical user would not consider all such combinations. In our sample distribution, on an average, out of the 93 sources/time-series, 10 are requested and it is representative of 335 queries, that were randomly generated. Even though the queries were chosen randomly, we ensured that each source is requested by at least one query. Figure 4 is representative of this query distribution. It is to be noted that the plot is with respect to the *size* of the query $|Q|$. It is also to be noted that

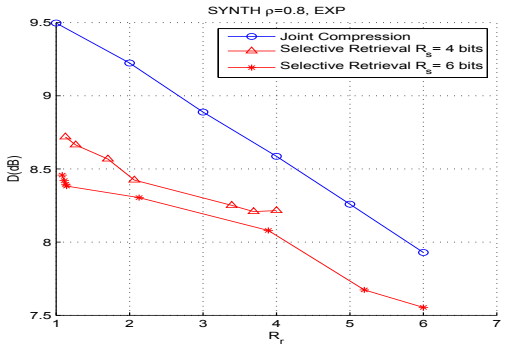

Fig. 6. Selective Retrieval vs. Joint Compression: SYNTH, $\rho = 0.8$

even a query set of size 335 *cannot be handled* with the naive storage technique presented in 2.1.2, i.e. compressing and storing every subset of sources separately, without paying an enormous price in total storage.

5.3. Performance Comparison

We compared the performance of joint compression and selective bit-retrieval for both the synthetic and real data-sets (see Figures 5, 6 and 7). The joint compression of the data set was performed by a Vector Quantizer (VQ) designed with the well known Generalized Lloyd Algorithm (GLA) [4]. The performance of proposed selective bit-retrieval technique was evaluated at two storage rates, $R_s = 4$ and $R_s = 6$ bits *per sample*. For the synthetic data-set with $\rho = 0.3$

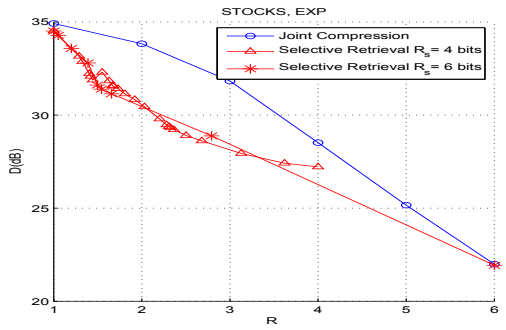

Fig. 7. Selective Retrieval vs. Joint Compression: STOCKS

(Figure 5), the selective bit-retrieval technique is able to provide a speed-up of nearly 5X at a distortion level of 9.35dB. For the synthetic data-set with $\rho = 0.8$ (Figure 6), there is a 4X speed-up over the joint compression technique, with average distortion of 8.5dB. Additionally, there is also a distortion gain of nearly 1dB at a retrieval rate of 1 bit per sample. In the real data-set (Figure 7), the selective retrieval technique provides a ≈1.6X speed-up with distortion 28dB and nearly 3.5dB less distortion at an average retrieval rate of 3 bits. We also note that increasing the storage rate generally results in better performance of the selective retrieval technique, except in the real data-set. This is because the design algorithm gets trapped in one the many local minima that riddle the cost surface.

6. CONCLUSIONS

We introduced the problem of fusion storage with selective retrieval for sensor/time-series databases. We posed the design problem as the minimization of a Lagrangian functional with an appropriate choice of an encoder, encoded bit-selector and decoder. We evaluated the necessary conditions for optimal solutions and proposed an algorithm that is guaranteed to converge to a locally optimal solution. We observed that the proposed algorithm is able to provide significant improvement in retrieval speed, for a given distortion level and significantly better data reproduction quality, for a given retrieval speed, over naive joint vector quantization, on both real and synthetic data-sets.

7. REFERENCES

[1] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side-information at the decoder," *IEEE Trans. on Information Theory*, vol. 22, pp. 1–11, 1976.

[2] Jayanth Nayak, Sharadh Ramaswamy, and Kenneth Rose, "Correlated source coding for fusion storage and selective retrieval," in *IEEE International Symposium on Information Theory*, 2005, pp. 92–96.

[3] R. Cristescu and M. Vetterli, "On the optimal density for real-time data gathering of spatio-temporal processes in sensor networks," in *Information Processing for Sensor Networks (IPSN)*, 2005, pp. 159–164.

[4] Allen Gersho and Robert M. Gray, "Vector quantization and signal compression," 1992, Kluwer Academic Publishers.